1990-03

Implementation of a Distributed Expert System for submarine shipboard maintenance using VP-Expert

Acton, David Wilber

Monterey, California. Naval Postgraduate School

(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

IMPLEMENTATION OF A DISTRIBUTED
EXPERT SYSTEM FOR SUBMARINE
SHIPBOARD MAINTENANCE USING *VP-EXPERT*

by

David Wilber Acton

March, 1990

Thesis Advisor: Tung Xuan Bui

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (*If applicable*) 55	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000	7b. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (*If applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*)	10 SOURCE OF FUNDING NUMBERS Program Element No / Project No / Task No / Work Unit Accession Number

11. TITLE (*Include Security Classification*)
IMPLEMENTATION OF A DISTRIBUTED EXPERT SYSTEM FOR SUBMARINE SHIPBOARD MAINTENANCE USING VP-EXPERT

12 PERSONAL AUTHOR(S) Acton, David Wilber

13a. TYPE OF REPORT	13b. TIME COVERED	14 DATE OF REPORT (*year, month, day*)	15 PAGE COUNT
Master's Thesis	From To	March 1990	71

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (*continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUBGROUP	Distributed Expert Systems, Knowledge bases, Meta-Expert System

19 ABSTRACT (*continue on reverse if necessary and identify by block number*)

Expert Systems (ES) are characterized by containing the knowledge of a single human expert. Most ES today operate on a "standalone" basis, providing expertise in a specific domain. However, managers making strategic decisions on complex topics require the coordinated assesment and evaluation of knowledge from multiple human experts. Standalone knowledge bases should be loosely or tightly coupled together to form a network of coordinated Distributed Expert Systems (DES). To facilitate this, a "meta-ES" could be designed to access and control these distributed knowledge bases, thus providing users with a single entry point into a vast knowledge network.

In the U.S. Navy submarine service, preventive maintenance is important for efficient operation. Since a submarine is constructed of compact, high energy systems, safety is paramount to prevent both personal injury and material damage during maintenance evolutions. The Ship's Duty Officer (SDO) is responsible for the safe and effective execution of all maintenance aboard ship. Thus, he needs to be knowledgeable of how maintenance on one system will affect the operation of other systems. Since the SDO requires many sources of expertise, automating a submarine shipboard maintenance process is an appropriate DES application.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Bui, Tung Xuan	22b TELEPHONE (*Include Area code*) (408) 646-2630	22c OFFICE SYMBOL AS/BD

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Approved for public release; distribution is unlimited.

Implementation of a Distributed
Expert System for Submarine
Shipboard Maintenance using *VP-Expert*

by

David Wilber Acton
Lieutenant, United States Navy
B.S., United States Naval Academy, 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the
NAVAL POSTGRADUATE SCHOOL
March 1990

Author: [redacted]

David Wilber Acton

Approved by: [redacted]

Associate Professor Tung Xuan Bui, Thesis Advisor

[redacted]

Assistant Professor Magdi Kamel, Second Reader

[redacted]

Professor David Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

Expert Systems (ES) are characterized by containing the knowledge of a single human expert. Most ES today operate in a "standalone" basis, providing expertise in a specific domain. However, managers making strategic decisions on complex topics require the coordinated assessment and evaluation of knowledge from multiple human experts. Standalone knowledge bases should be loosely or tightly coupled together to form a network of coordinated Distributed Expert Systems (DES). To facilitate this, a "meta-ES" could be designed to access and control these distributed knowledge bases, thus providing users with a single entry point into a vast knowledge network.

In the U.S. Navy submarine service, preventive maintenance is important for efficient operation. Since a submarine is constructed of compact, high energy systems, safety is paramount to prevent both personal injury and material damage during maintenance evolutions. The Ship's Duty Officer (SDO) is responsible for the safe and effective execution of all maintenance aboard ship. Thus, he needs to be knowledgeable of how maintenance on one system will affect the operation of other systems. Since the SDO requires many sources of expertise, automating a submarine shipboard maintenance process is an appropriate DES application.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

Expert Systems (ES) are characterized by containing the knowledge of a single human expert. Most ES today operate in a "standalone" basis, providing expertise in a specific domain. However, managers making strategic decisions on complex topics require the coordinated assessment and evaluation of knowledge from multiple human experts. To facilitate using a DES, a "meta-ES" could be designed to access and control the distributed knowledge bases, thus providing users with a single entry point into a vast knowledge network. This thesis explores new developments in Information Systems (IS) technology in the area of Distributed Expert Systems (DES).

The study of DES can lead to important practical implications. In the U.S. Navy submarine service, preventive maintenance is important for efficient operation. Since a submarine is constructed of compact, high energy systems, safety is paramount to prevent both personal injury and material damage during maintenance evolutions. The Ship's Duty Officer (SDO) is responsible for the safe and effective execution of all maintenance aboard ship. Thus, he needs to be knowledgeable of how maintenance on one system will affect the operation of other systems. Since the SDO requires many sources of expertise, automating a submarine shipboard maintenance process is an appropriate DES application.

B. OBJECTIVES

This thesis determines the feasibility of building a DES by researching current DES technology and prototyping a submarine shipboard maintenance system using *VP-Expert*, a commercial off-the-shelf ES software package.

C. RESEARCH QUESTIONS

1. Is a meta-ES application to control a DES feasible?
2. Is submarine shipboard maintenance planning an application which would benefit from a DES?
3. Will *VP-Expert* be an adequate shell for building both a meta-ES and a DES?
4. How will the separate knowledge bases comprising the DES be coupled?

D. SCOPE, LIMITATIONS, AND ASSUMPTIONS

This thesis focuses on the DES technology and the feasibility study of building a DES application using *VP-Expert* software. DES is a relatively new technology and the author hopes that this thesis will shed some light on this important area of the IS industry. This research effort concentrates on the implementation of a submarine maintenance DES prototype. The author assumes that readers of this thesis have some IS background in the areas of ES and database management and have some familiarity with ES shell programs.

E. METHODOLOGY

The methodology used in this thesis is twofold. First, a survey of researching DES technology was conducted and second, a DES prototype was implemented using *VP-Expert* software to validate the proposed techniques for DES architectures.

F. CHAPTER OUTLINE

The thesis is organized as follows. Chapter 2 surveys current distributed knowledge base technology. It explores the present status of DES in the IS industry, compares DES with distributed processing and discusses communication and design issues associated with DES. Chapter 3 presents a generic framework for a DES model, discusses the components and IS applications contained in a DES and investigates coupling issues associated with distributed knowledge bases. Chapter 4 presents a working prototype of a DES application: the Submarine Maintenance Expert. It demonstrates how the system could be used for an actual consultation, discusses lessons learned using *VP-Expert* and *dBASEIV* software and provides directions for further research. A sample consultation of the DES prototype is presented in Appendix A. The *VP-Expert* programs for the prototype are listed in Appendix B.

II. A SURVEY OF DISTRIBUTED KNOWLEDGE BASE TECHNOLOGY

A. OVERVIEW

As mentioned in the introduction, standalone Expert Systems (ES) can be pooled together to form a Distributed Expert System (DES) thus allowing managers to make strategic decisions on complex topics. In this paper, the term "knowledge base" is used to define the source of expert information necessary to solve a given problem. Thus, a "knowledge base" can be an ES, the detached information base (data base) supporting an ES, or an on-line human expert. A "meta-ES" could be designed to access and control these distributed knowledge bases, thus providing a user with a holistic view of the complete cross-functional management process. [Ref. 1]

B. COMPARISON BETWEEN DISTRIBUTED KNOWLEDGE BASES AND DISTRIBUTED PROCESSING

Although both distributed knowledge bases and distributed processing involve the integration of computer resources, distributed processing is an exceedingly complex subject. According to Kroenke, distributed processing is still in its infancy, is constantly evolving, and will continue to change as new developments occur during the next several years [Ref. 2]. It is briefly discussed here to show how it compares, and does not compare, with distributed knowledge bases.

1. Distributed Processing

The standard characteristics of distributed processing is the use of computer processors at geographically separate locations connected by data communication channels to share data and computer resources [Ref. 3]. Improved semiconductor technology has lowered the processing costs of microcomputers, and since it is economically advantageous to use the least powerful computer capable of performing a given processing task, distributed processing via Local Area Networks (LANS) is becoming ever more popular [Ref. 3]. The sharing of data, applications, and communication through electronic mail all result in a substantial benefit to a larger user community. Sharing also allows larger, more cost-effective data storage devices to be used [Ref. 4].

Distributed systems can be created by decentralizing existing computer systems, by connecting formerly separate systems, or by creating an entirely new system. The nodes of a distributed computer system can be any type of computer from mainframe to micro, or a peripheral such as printers, plotters, modems, external storage, or any other form of computing hardware. The hardware at each node can be selected for its appropriateness to a given application, or every node can perform the same function. In the latter case, the reliability of the distributed system can increase if each node can cover the functions of any failed node. [Ref. 4]

Whereas the managers of mainframe systems can rely on many years of experience, the management of distributed computer systems is still a relatively new

and evolving field. Managing a distributed system is more complex due to potential security problems, concurrency, failure/recovery and system control.

2. Distributed Knowledge Bases

In its most basic form, a distributed knowledge base is simply a collection of specific, partially related knowledge bases. The concept here is to give a decision maker a single access point to multiple knowledge bases. This can be accomplished by designing a meta-ES which utilizes several knowledge bases to solve complex managerial problems necessitating complex compilation of multiple aspects and analyses of the problems. Unlike distributed processing, distributed knowledge bases need not be geographically separated, in fact, they may be a collection of knowledge bases residing in the same data storage device.

C. ARCHITECTURE OF DISTRIBUTED KNOWLEDGE BASES

The key to a successful meta-ES is its ability to effectively communicate with multiple knowledge bases. Managers need access to knowledge from different areas of expertise to make productive decisions. According to Bui:

> ...the absence of communications between specific systems as well as their inability to deal with uncertainties caused by ad hoc changes during departmental decision making processes often result in serious conflicts among decisions and implementing strategies [Ref. 1].
>
> However,...the analysis and design of communications support should go beyond the usual focus on technical issues of communications control such as network topology, network design, capacity and flow assignment, error detection, and so on. [Ref. 5]

Although cross-communication is important, a loosely-coupled architecture for the meta-ES is equally important to ensure autonomy of the subsystems. As defined by Page-Jones, "coupling" is:

> the degree of dependence of one module on another; specifically, a measure of the chance that a defect in one module will appear as a defect in the other, or the chance that a change to one module will necessitate a change to the other [Ref. 6].

One way to measure coupling is by the degree of interdependence between two modules. Low (loose) coupling between modules indicates a well partitioned system in which the modules are as independent as possible. Conversely, a tightly coupled system is usually characterized by the "relationships" between modules. Some of these relationships are unnecessary, too numerous or both. Each module within the system must worry about the particular internal construction details of any other. [Ref. 6]

Probably the most important property for supporting a flexible (loosely-coupled distributed) system is that of communication transparency in which the same communication primitives are provided for remote and local transactions [Ref. 7]. Additionally, strict controls at the "coordination" level is needed to reduce miscommunications within the organization [Ref. 1].

For a DES, the ES shell program must tie the distributed knowledge bases together. In this function, the ES shell can be thought of as a "system server." Servers interact with users in a transparent fashion, just the same way as users would interact with other users. The ES shell language "must support safe, convenient communication for a dynamically changing mix of loosely coupled processes--

processes designed in isolation, and compiled and loaded at disparate times" [Ref. 8]. Therefore, the ES shell must interact with its environment through messages, in a similar manner as Interprocess Communications (IPC) interact with distributed operating systems. For an ES shell to properly interact with a DES, it needs the attributes similar to IPC attributes. IPC are important for the smooth and safe utilization of distributed knowledge bases, but are quite complex in their interaction with the shell's Operating System (OS). Scott lists three reasons for the complexity of IPC: convenience and safety, error handling and protection, and concurrent conversations.

1. Convenience and Safety

IPC is more structured than OS file operations. The shell's request to the distributed knowledge bases resemble procedure calls more than they resemble the transfer of uninterpreted streams of bytes. IPC transfer arbitrary collections of *program variables without sacrificing type checking, and without explicitly packing* and unpacking buffers.

2. Error Handling and Protection

IPC is more error prone than OS file operations. Hardware and software can fail. Unlike OS files, IPC's display much more nondeterministic behavior. Fault-tolerant algorithms may allow the shell to recover from many kinds of failures. The shell must not be vulnerable to erroneous behavior on the part of the users. However, the algorithms must also be loosely coupled. Errors in communication with any particular knowledge base should not effect the access of others.

3. Concurrent Conversations

While a conventional sequential program typically has nothing interesting to do while waiting for a file operation to complete, a distributed knowledge base shell usually does have other work to do. Efficiency and clarity may best be realized with a dynamic set of tasks within a shell, one for each uncompleted request. [Ref. 8]

D. DESIGN ISSUES

In designing a distributed knowledge base, Bui devised three strategies that can be used in the distribution: centralized knowledge base, hierarchical knowledge bases, and partitioned knowledge bases.

1. Centralized Knowledge Base

Under this strategy, all the knowledge is pooled into a central single data storage device. This minimizes networking problems due to incompatible operating systems, and provides for the most effective control of the knowledge base. Only one copy of the various subsets of the knowledge base needs to be kept. Redundancy and contradictions among the rules stored in the knowledge base can be easily identified and quickly resolved due to the entire knowledge base being in one location. However, the centralization strategy can involve higher communication costs if the central knowledge base is accessed continuously for each consulting session from geographically separate user locations.

2. Hierarchical Knowledge Bases

In this strategy, knowledge is layered in various levels of abstraction from the highest level meta-knowledge about the various aspects of the knowledge

distribution to the lowest level detailed and intensive knowledge for specific ES. The knowledge bases can also be layered in terms of their geographic characteristics and application. This may result in the different knowledge bases having a certain degree of dependence on each other. The meta-knowledge forms the core expert that can intervene and guide the execution of the rest of the knowledge bases whenever required.

3. Partitioned Knowledge Bases

Here each knowledge base, although part of a single distributed knowledge base network, is inherently independent. The advantage of this strategy is that each knowledge base is loosely coupled and each ES application can solve most of the problems that occur during a consultation in their own locations without having to access the meta-expert system in the network. The most difficult part of this strategy is minimizing the transactional flags between each knowledge base such that each knowledge base can operate both independently and as a subset of the distributed knowledge base network. [Ref. 1]

E. SUMMARY

Distributed knowledge bases is a new technology in the information systems industry. The ability for managers to utilize multiple knowledge bases during a single ES consultation should greatly improve strategic decision making involving complex issues. However, as with any new technology, there are some problems to overcome to improve the benefit of distributed knowledge bases. These problems involve the coupling strategies used to tie the separate knowledge bases into one distributed network without sacrificing each individual knowledge base's indepen-

dence. The framework for a meta-ES which attempts to overcome these problems will be discussed in the next chapter.

III. A FRAMEWORK FOR A DISTRIBUTED EXPERT SYSTEM

A. OVERVIEW

Chapter 2 addressed some new developments in distributed knowledge base technology, which allow users to make strategic decisions requiring coordinated assessment and evaluation of multiple managerial expertise. In this chapter, a distributed ES (DES) will be developed which will utilize this new distributed knowledge base technology.

B. THE DISTRIBUTED EXPERT SYSTEM MODEL

Figure 3-1 shows the framework of a DES. The network consists of several independent information systems (IS) applications, connected by a "meta-ES." The meta-ES provides overall control of the DES network. This network scheme is similar to a star topology. Here, the meta-ES is the central element which links the individual ES. Unlike a true "network" however, the meta-ES does not establish a dedicated path between several users running individual ES applications and wishing to communicate. Rather, the meta-ES provides a "Top-Down" hierarchical approach for the user to view the network. The user only has to access one application--the meta-ES--to access the entire distributed knowledge base. Thus, the user effectively has the input of several human experts to assist him/her in solving a complex problem, without needing to know beforehand which specific ES to call to solve the problem. The meta-ES makes those decisions. The components which comprise a

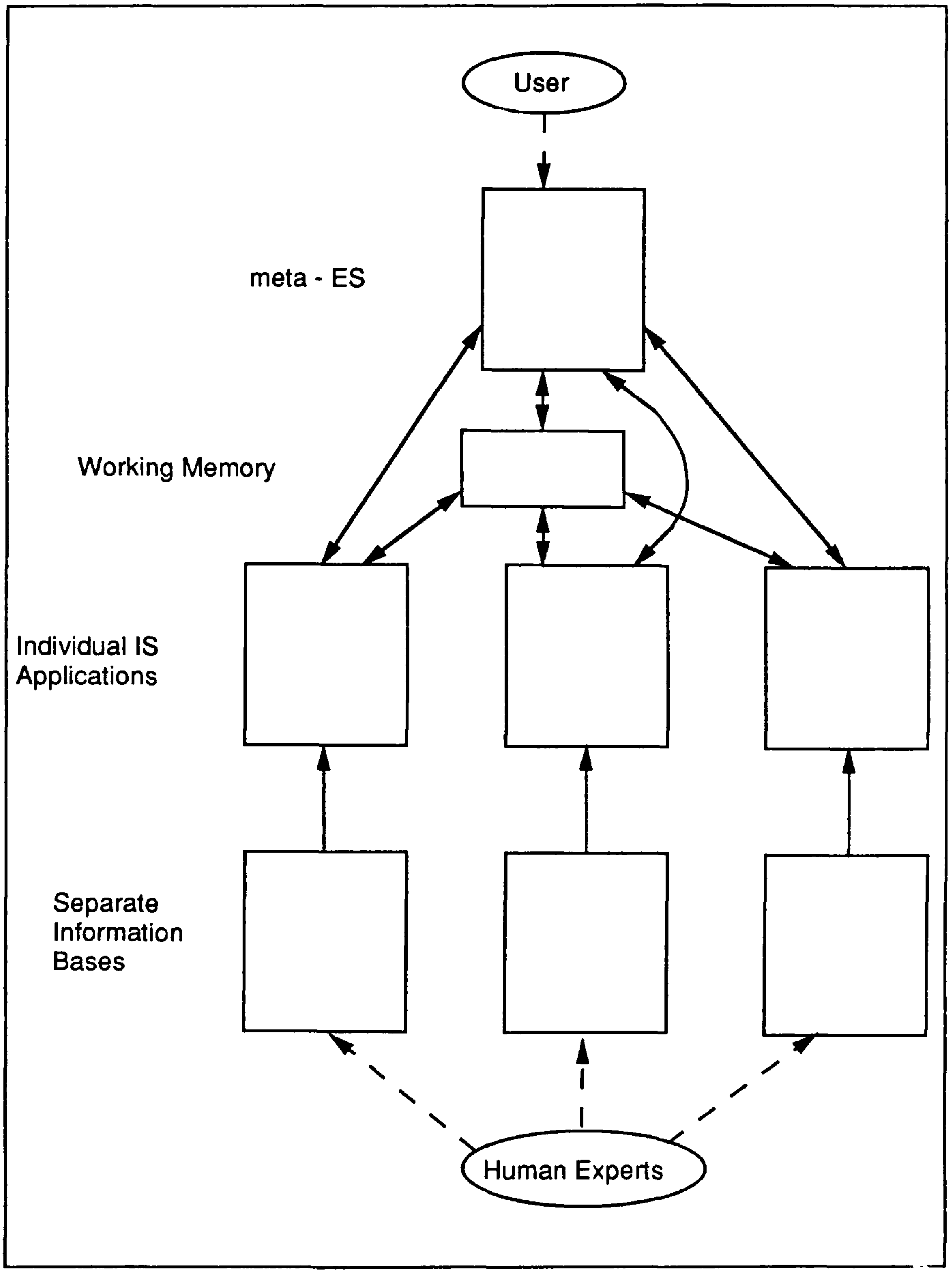

Figure 3-1 A Model for a Distributed Expert System

DES are the individual IS applications, the meta-ES and its associated working memory. The individual IS applications may also utilize separate information bases.

1. Individual Information Systems Applications

To solve complex problems, the DES contains several independent, yet related IS applications. The applications need not be all ES, rather they can be a combination of database management, decision support, and ES programs. They need to be related in the sense that they all contribute to a manager's ability to make decisions for a given complex topic.

The combination of several IS applications allow a decision maker to draw information from a broad spectrum of relevant sources. A database application can be utilized to maintain a list of available ES topics covered by the DES network. Each DES session could start by accessing the database application first to allow the user to simply select an available ES application from the database. With the ES application known, a decision support application could by called to begin collecting data from the user to determine what type of decisions need to be made and what ES applications will need to be utilized to help the user make the best decisions. With specific data obtained, one or more ES applications would then be sequentially accessed to provide the comprehensive and coordinated expertise required to make complex and strategic decisions. The expert knowledge associated with each ES application can either be an internal set of rules or a separate data base file called an "information base." The advantages of separate information bases will be discussed later.

2. The Meta-Expert System

As stated earlier, the meta-ES is the heart of a DES and provides overall control of the network. The meta-ES is the shell which identifies the problem, decomposes the problem, devolves problem solving to a specific ES application in the network, and synthesizes solutions [Ref. 9].

Depending on the complexity of the DES network, the meta-ES itself may be a simple program which provides directory services to other network applications, or it may be a sophisticated ES providing coordinated assessment of each individual ES application's output. The meta-ES must know the domain of the network. It must know the area of expertise and problem-solving goal of each application. The meta-ES controls the coupling required by each application. It must know the data structures used by each application to allow smooth transitions and type compatibility between them.

The meta-ES controls both the flow of data and the sequential operations of the network. As an example, the meta-ES may start each session with a brief introduction of the DES capabilities. Then it may guide the user to select a topic covered by the network. This may be accomplished using a database application as previously discussed, or it may simply be a checklist inside the meta-ES. Given a topic, the meta-ES would then activate a specific ES application capable of assisting the user. This is done by saving into working memory the data obtained from the initial consultation. The called application would first access the working memory and determine the status of applicable transitional flags (discussed later) and relationships with data in its own information base. The consultation would

continue, updating the working memory as it guides the user to decisions using that application's inference engine and information base. If one ES application is insufficient to solve a complex problem, the meta-ES would again save to working memory all data obtained from the previous consultation and call on an additional ES application, and so on until the user is satisfied and/or the network is exhausted. The meta-ES may even be able to recommend additional DES networks for the user based on specific problem domains.

3. Working Memory

The working memory associated with the meta-ES is necessary as a data storage buffer. This working memory may be a separate text file or internal to the meta-ES application. The individual applications may have their own information bases or their own set of rules to obtain data/input from the user. To minimize tight coupling requirements and to maintain data integrity, the individual information bases should never be updated by the meta-ES as a result of a consultation. Rather, all data used/generated during a consultation is stored in the working memory for the duration of the session. Every IS application in the network has access to the working memory and their own information base, but never to the information base of a different application. Each individual application may have the need to update its own information base from the data stored in the working memory. That decision will be made by the user and the application itself, not the meta-ES.

Transitional flags need to be utilized to act as pointers during consultations. The status of these flags is also stored in the working memory. These flags are used to keep track of which applications have been run and also monitor the

status of any on-line application. These flags are checked before and after each application is called. This allows for smooth transitions between applications and can improve understandability, clarity, and user friendliness. For instance, each application can have an introduction screen displayed when it is called. But if the application is called repeatedly, the user would soon get tired of viewing the introduction screen over and over again. Instead, a transitional flag can be marked such that the introduction screen is only displayed the first time an application is run.

After a session is complete, the working memory is cleared of all data and transitional flags so that follow on consultations start anew.

C. COUPLING REQUIREMENTS IN A PARTITIONED DES

There are minimal relationships which must exist for a DES to function. However, any relationship implies some form of coupling restriction between individual applications in the DES network.

As mentioned in Chapter 2, a goal of distributed knowledge bases is loose coupling. This ensures that each application can run on a standalone basis for very specific consultations, or also be run as part of the DES network. Also, this allows each individual application to be updated and/or changed periodically without having to change any portions of other applications in the network. To achieve minimal coupling, each ES application should be designed to operate independently, and the meta-ES should be designed to connect the network together. The designers of the IS applications contained in the DES need to view their designs from a Bottom-Up perspective as shown in Figure 3-1. The designers must take into

consideration the effect of updating the individual information bases during a user's consultation, and the effect of altering the basic structure of each information base. The *records* of an information base can be updated (added, modified, or deleted) without any effect on the performance of the IS applications, but serious coupling problems arise if *fields* are updated. This is due to the coded commands within the ES applications. The applications can call on specific information bases regardless of size, and access any or all records in each information base. However, the applications also call specific fields within each information base. Therefore, if any of these fields are updated, then the actual ES application code must also be updated.

1. Coupling Design Issues

Two or more applications are said to be *common coupled* if they refer to the same global data area [Ref. 6]. This is normally a poor practice since one application's information base could be advertently or inadvertently changed by a different application. Also, every information base call must use common, explicit field names. To avoid some of the problems associated with common coupling when designing a DES, each ES application should store its expert's knowledge in a separate information base. This protects each application's information base from being changed by the actions of a different application. Additionally, this allows the information base of each ES application to grow without requiring any changes to the basic inference engine structure. However, this does not remove the important issue of explicit field names used in the information bases. In order for each application to function properly with data passed to it from working memory, the

field names used by each individual information base must be identical. If the field names in each information base are unique, then data sharing between applications would be very difficult and the design of the meta-ES would be extremely complex. The meta-ES would need great artificial intelligence capability to draw inferences and set relations based the *value* of the data, vice simply comparing field names, in order to pull the appropriate records from the information bases.

The meta-ES must be designed last since it needs to know what the problem solving domain of the network is and also what individual applications will comprise the network. It should become apparent that the complexity of the meta-ES design is inversely related to the degree of coupling between the applications within the network. A very tightly coupled system can be controlled by a simple meta-ES which only provides directory services. However, an ideal loosely coupled network will require a very complicated meta-ES capable of relating attributes between unrelated information bases and drawing its own inferences based on data obtained from each individual ES consultation. If the information bases for the individual applications have related foreign key fields[1], then the meta-ES can act simply as a directory between applications. Data obtained during one session is simply forwarded to the next application where the information base is accessed for records with the same field names. If, however, no field names are the same within individual information bases, then the meta-ES is tasked with assigning temporary variables to the attributes of the records used in each application. Those variables

---

[1] A "key" is a group of one or more fields which uniquely identify a record in a database. A "foreign key" is a common field name between database files and is a key of a different relation. [Ref. 2]

are compared against the information base of the next application to be run, to see if there are any field name matches. If not, the meta-ES must be able to relate each attribute from one application with the proper attribute in the next application, each of which will have different names.

D. SUMMARY

A meta-ES is the application developed to provide the overall control of a DES network. From the user's Top-Down view of the DES architecture, the meta-ES provides a single access point to the entire DES. The degree of coupling which exists between the individual applications in the network determines the design complexity of the meta-ES. From the Bottom-Up perspective of the information base designers, rigid data field standards provide minimum coupling requirements. It was shown that the degree of coupling and the complexity of the meta-ES design is inversely related. Therefore the goal of having a loosely coupled system requires a very complicated meta-ES application to control it.

IV. A DES APPLICATION: THE SUBMARINE MAINTENANCE EXPERT

A. OVERVIEW

In the U.S. Navy submarine service, preventive maintenance is important to keep the boats operating at peak efficiency. Since a submarine is a relatively small weapons platform requiring great capabilities, it is constructed of compact, high energy systems. These systems are located throughout the ship as space permits. Although most systems are independent, maintenance on any one system may adversely affect neighboring spaces. For example, portions of one system may have to be removed to physically access another system, maintenance on portions of a piping systems may require isolation of the entire system, etc. Also, due to the high energy capacity of many systems, safety is paramount to prevent both personal injury and material damage during maintenance evolutions.

These are some of the many complexities involved when coordinating shipboard maintenance. The Ship's Duty Officer (SDO) is responsible for the safe and effective execution of all maintenance aboard ship. As a result, he needs to be knowledgeable of all the relationships between systems on board and how maintenance on one system will affect the operation of other systems. Additionally, he needs to know what applicable safety precautions are required for each specific maintenance task. To get this knowledge, the SDO relies on senior Petty Officers, other Department Heads, and numerous maintenance and safety publications. He also has his own experience to draw on. All this results in long, detailed reviews of

every maintenance task, to determine exactly how it will affect other departments on the ship and to ensure that all applicable safety precautions are followed.

Thus, the SDO routinely needs the advice of several human experts. For the SDO to access this advice requires that the human experts be available around the clock. As an alternative, each human expert could contribute to the construction of an ES to assist the SDO in solving complex problems, but this would result in several individual ES applications. The SDO would need to know beforehand which ES application to access in order to obtain certain information. Therefore, a DES containing all the appropriate ES applications should alleviated these problems and lend itself perfectly for submarine shipboard maintenance advice.

B. A FRAMEWORK FOR THE SUBMARINE MAINTENANCE DES

Figure 4-1 shows the framework used by the Submarine Maintenance DES. The system is comprised of two ES applications, one Database Management (DBMS) application, one meta-ES controlling the network and a simple text file working memory. The DES uses a centralized knowledge base configuration (as discussed in Chapter 2) and each application has its own separate information base.

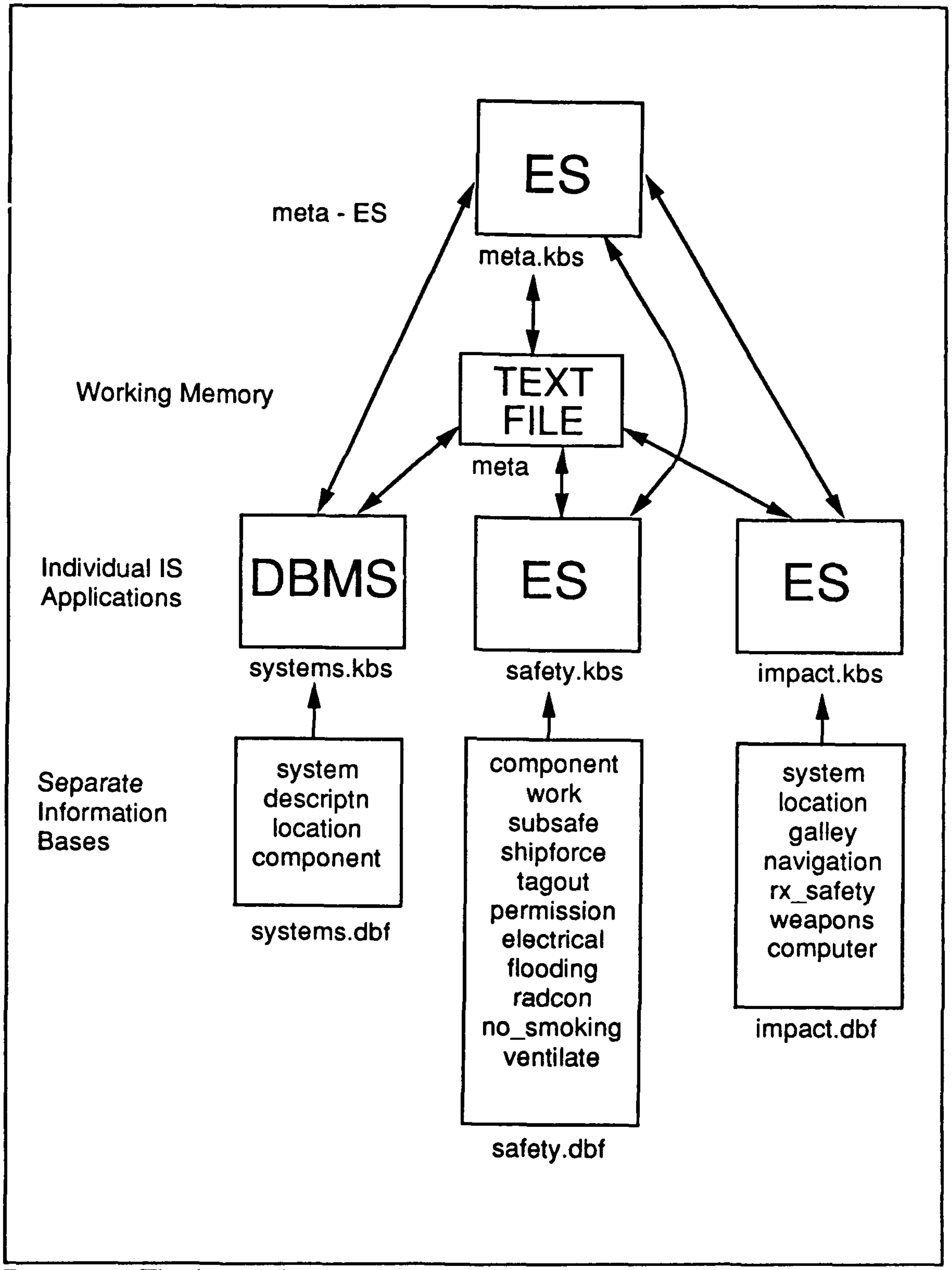

Figure 4-1 The Submarine Maintenance Distributed Expert System

All applications were coded using *VP-Expert* software[2]. The information bases were created using *dBASEIV* software. For the two ES applications, the information bases contain their respective expert's knowledge. For the DBMS application, the information base contains simple database records. All applications have access to the text file "META" which acts as the working memory associated with the meta-ES. As discussed in Chapter 3, there is some minimum degree of coupling required by the information bases in these applications to minimize the complexity of the meta-ES. The requirements are that the related field names in each information base are identical. This allows easy location of pertinent data in the separate knowledge bases when called by the meta-ES.

An iterative design methodology was used to generate all the applications used in this project. The author (designer) has experience as a maintenance officer aboard a U.S. Navy submarine and found this methodology to be quite efficient and effective. The intent of this project was to show *VP-Expert's* potential as a shell to design a DES, not to satisfy any U.S. Navy requirements for an actual shipboard application. Thus, the ES applications contained in this DES have fairly shallow information bases and inference engines. The inference strategy used by each application is modus ponens, as this is the strategy used by *VP-Expert*.

Any user of this project must be familiar with *VP-Expert* as the error handling routines all default to *VP-Expert's* shell introduction and control screen. The user must be able to navigate through this control screen. Refer to Appendix A for

---

[2] For simplicity, the DBMS application was written using *VP-Expert*, however, a larger scale project would benefit by coding the DBMS application using DBMS software for greater program efficiency.

screen displays and options available during an actual consultation with the Submarine Maintenance Expert, and to Appendix B for the *VP-Expert* code for each application.

1. Systems Database Management Application

The first application in the Submarine Maintenance DES called by the meta-ES is a simple DBMS application "SYSTEMS.KBS" which allows the user to select the exact component/system to undergo maintenance. There are literally hundreds of thousands of specific components on the submarine. To avoid the problem of lengthy information base searches each time the user typed in the name of a component, and to avoid the problems associated with character string mismatches during data entry, the DBMS application is menu driven. As shown in Figure 4-1, the information base "SYSTEMS.DBF" contains four fields: SYSTEM, DESCRIPTN, LOCATION and COMPONENT. Each menu has limited choices and progresses through the information base in a logical method to find the specific component to undergo maintenance.

First, the program displays a brief introduction and then a screen listing all the different systems on board the ship. The user selects the system containing the component. Next, the program displays a screen containing a generic description (such as valve, pump, circuit breaker, etc.) of each type of component contained in that selected system. The user selects the generic category best describing the component. At this point, to avoid choosing among numerous specific components (such as "valves," since some systems have thousands of valve-type components), the program next displays a screen showing the different spaces (compartments) aboard

ship which contain the specified system. The user then selects the appropriate location. Finally, the program displays a list of all the specified components located in the selected space, and the user selects the exact component which will undergo maintenance. The DBMS application saves to working memory all the data gathered during the session, and returns to the meta-ES.

2. Safety Expert System Application

The next application called by the meta-ES is a safety ES "SAFETY.KBS." Given the component to undergo maintenance and the type of work to be performed on the component, the ES then determines several things: the degree of work control, the permission requirements to approve tagouts and to start work, and the appropriate safety precautions. The safety ES uses an information base "SAFETY.DBF" containing the following eleven fields: COMPONENT, WORK, SUBSAFE, SHIPFORCE, TAGOUT, PERMISSION, ELECTRICAL, FLOODING, RADCON, NO_SMOKING and VENTILATE.

When the safety ES is called by the meta-ES, it first retrieves from working memory the name of the component to undergo maintenance. The user is then presented a choice of possible work options to be performed on that component. The user selects the appropriate work. Then the program determines if the work is subsafe[3] or not, and if shipforce personnel are capable of performing the maintenance. Some systems on the ship are inherently more dangerous than

[3] "Subsafe" is a code assigned to specific components which are critical to the watertight integrity of the submarine. Subsafe components require the most demanding in-process work controls and stringent post-maintenance retests to ensure strict compliance with submarine safety doctrine.

others, so strict controls are placed on tagging "out-of-service" certain portions of the system prior to beginning maintenance. Different levels of permission (CO, Engineer, etc.) are required for both the tagouts and to actually start work. The safety ES will determine whose permission is required for each of these. Next, the safety ES will determine which safety precautions are applicable to ensure that no personal injury and no equipment damage occurs during the maintenance effort. The application then returns to the meta-ES.

3. Impact Expert System Application

The last application called by the meta-ES is a maintenance shipwide impact ES "IMPACT.KBS." Given the location on the ship and the system undergoing maintenance, the program determines if other departments will be affected. If so, the application lists which other department heads should be notified and explains why it is important to notify them. The impact ES uses an information base "IMPACT.DBF" *containing the following seven* fields: SYSTEM, LOCATION, GALLEY, NAVIGATION, RX_SAFETY, WEAPONS, and COMPUTER.

When the impact ES is called by the meta-ES, it first retrieves from working memory the name of the system and the location of the work area. Since the duration of the maintenance is critical for scheduling purposes, the program gets a job duration estimate from the user by presenting a choice of possible time periods. The user selects the time period which most closely reflects his maintenance duration estimate. The program then determines which departments will

be affected by the maintenance and what actions are necessary to minimize the impact on ship's operations.

C. LESSONS LEARNED USING *VP-EXPERT* (VERSION 2.02)

Several ES development tools exist in the marketplace today. *VP-Expert* was chosen to develop the Submarine Maintenance DES for several reasons. First, *VP-Expert* was relatively inexpensive ($39 for the student version) and easy to obtain. However, *VP-Expert* was also extremely simple to use yet powerful in its ES capabilities. *VP-Expert* proved to be very well suited for designing and implementing a DES due to its knowledge base chaining functions. Only minimal coupling was required to combine several standalone ES applications into a functional DES controlled by a single meta-ES application.

1. Learning Curve

Using the tutorial, a student with no prior ES training can learn to use *VP-Expert* effectively in under twelve hours. The student tutorial covers all the major functions and capabilities of *VP-Expert* simply and without annoying redundancies or errors. The tutorial is concise and logically leads the student through the basic construction of ES and gradually builds a relatively complex, chained ES through use of explicit examples.

2. Nested Loops

The use of whiletrue-then loops or whileknown-then loops is a tremendous programming advantage and saves much coding for repetitive actions in programs. Additionally, loops inside of loops (nested loops) allow for even greater programming efficiency by minimizing the code required to perform numerous

repetitive tasks. In *VP-Expert*, loops are terminated using an "END" statement. Unfortunately, a single END statement terminates <u>all</u> loops above it. Therefore, nested loops are not possible in *VP-Expert*. This was the source of great frustration during early programming attempts. Once this drawback was discovered, work-arounds were fairly simple and did not limit the effectiveness of *VP-Expert*.

3. Resetting Variables

When chaining multiple ES together, data and variable values are first stored in working memory using the SAVEFACTS command. The stored data is then later retrieved by the called (chained) ES by using the LOADFACTS command to continue the consultation. Since *VP-Expert* utilizes backward chaining and depth first searching strategies in its inference engine, if a variable value is not UN-KNOWN, then the inference engine will default to the first rules containing those assigned variables. Also, due to *VP-Expert* using monotonic reasoning, it will not attempt to reassign a value to a known variable, even if the variable value is irrelevant for the current consultation. These restrictions can result in a functionally correct ES locking up in an infinite loop, never asking for user input, or a dead ES which simply defaults to the first rule in its information base every consultation.

The workaround to overcome these problems is actually quite simple. By using the RESET command before every FIND command, all variable values will be UNKNOWN and the rule base will fire correctly. This greatly relieves a major coupling necessity by not requiring variables in distributed ES to have exact, limited values allowed to be assigned to them. Another option to minimize irrelevant data passing after each session is to first RESET ALL variables. Then pertinent variables

can be assigned specific values before using the SAVEFACTS command. Otherwise the SAVEFACTS command will save to working memory the value of every variable used by any portion of the ES, resulting in extraneous data capture.

4. Resetting Pointers

VP-Expert uses pointers to keep track of the records in external information bases. During searches (initiated by the GET command), each record in the information base is looked at only once, and the pointer is incremented to the next record. This may result in several "lost" rules on future calls to that information base. The information base becomes effectively smaller after each program execution. The workaround here is also quite simple and is similar to resetting variables prior to each FIND. The CLOSE command must be used just prior to each GET command to reset the information base pointer to the first record. This action ensures that the entire information base is available for each consultation.

5. Path Limitations

VP-Expert allows path statements to be used so ES applications can be stored in directories separate from the *VP-Expert* shell programs. Also, information bases stored in separate directories can also be called using a path statement in the ES application. The path command also works well in using *VP-Expert* to design a DES since this allows the individual ES applications to be distributed. However, this does create a minimal coupling requirement. Since all *VP-Expert* applications default to the directory containing the shell programs, the individual ES applications must use path statements when calling their own subprograms (such as calling separate information bases). For example, ES application #1 and its separate information

base reside in directory X. When run on its own, ES application #1 could call on its information base without a path statement since they both reside in the same directory. But if ES application #1 is called by a meta-ES from directory Y, then directory Y becomes the default directory. When ES application #1 calls its information base, it must use a path statement to access directory X, or else the call will look unsuccessfully for the information base back in directory Y, and a runtime error will result.

The workaround here is simple if some coupling requirements are set. Before every call in every application, use a path statement to the directory containing the called file/program. However, if any files/programs contained in the DES are moved between directories, then both the moved files and the meta-ES need to be updated with correct path statements.

6. *dBASE/VP-Expert* Boolean Compatibility

When entering boolean values into a *dBASE* file, either Y or T is accepted for TRUE values, and either N or F is accepted for FALSE values. However, on the computer screen, only T or F is displayed regardless of what was entered by the user. The rules in *VP-Expert* check variable values by exactly matching character strings. Thus, if a rule was premised with "IF variable = Y" but the information base variable equaled "T," the premise would be incorrectly evaluated as false and the rule would not fire. Therefore, in order to make the *VP-Expert* information base rules fire correctly, all boolean expressions must be written inclusively in the form: "IF variable = Y OR variable = T (or "N OR F")." This was another source of great frustration in debugging logically correct ES applications.

D. SUMMARY

The Submarine Shipboard Maintenance DES proved to be a functional DES project. Using *VP-Expert*, all applications were constructed using an iterative design methodology resulting in both efficient and effective programs. Only minimal coupling constraints were required to combine the individual applications into the network, resulting in a fairly simply designed meta-ES. DES technology should blossom with the advent of simple to use ES shells such as *VP-Expert*, which requires only minimal overhead to learn its great potential. The few problem areas discovered during coding with *VP-Expert* were quickly solved and workaround routines were possible.

APPENDIX A

I. A SUBMARINE MAINTENANCE EXPERT CONSULTATION

A. OVERVIEW

This appendix is an example of a consultation using The Submarine Maintenance Expert. The figures which follow are similar to actual screen output of the consultation as run on an IBM AT personal computer. In *VP-Expert*, the user highlights menu items to be selected. In this appendix, highlighted items are shown in **bold and underlined** type. Generally, the user first accesses the meta program, which controls the three applications representing the distributed expert system as described in the body of this thesis. From the meta program, the user selects the component to undergo maintenance, then finds the applicable requirements and safety precautions, and then finds out what impact the maintenance will have on other shipboard departments. The user can run each application separately and/or repeatedly as he desires. The total run-time of a typical session is between two and three minutes.

B. THE CONSULTATION

Figure A1 is the opening screen presented by *VP-Expert* when executed from DOS. To begin any consultation, the user selects option 4, "Consult."

Each application written in *VP-Expert* is categorized as a "knowledge base." Figure A2 lists all the knowledge bases which reside in the same DOS directory as

the *VP-Expert* main program (if the knowledge bases resided in a different directory, then option 7, "Path" of Figure A1 would need to be selected prior to option 4, "Consult"). For the Submarine Maintenance Expert, "META" is selected since it is the application which controls the distributed Expert System.

Figures A3 and A4 simply presents the author's introduction screens for The Submarine Maintenance Expert. These two screens will be displayed only once during each session. This promotes program efficiency by not requiring the user to view them repeatedly during multiple consultations.

Figure A5 presents the user with the preferred order to utilize the Submarine Maintenance Expert. Option 1, "Select Component" is chosen to begin the consultation. This screen will reappear after each individual application is completed.

Figure A6 is simply an introduction screen for the Maintenance Database program. As with the other introduction screens, it will only be presented once the first time this application is run.

Figures A7 through A11 guide the user in selecting the exact component which will undergo maintenance. In this example, the Trim and Drain system is first selected. Next, the type of component, tank, is chosen. The tank is located in the Torpedo Room, and specifically it is the Auxiliary tank #2. This "walkthrough" to find the exact component mimics the thought process the Expert uses. Finally, the application allows the user to confirm his choice, or run the application again to select a different component. In this example, Auxiliary tank #2 is correct, and a screen similar to Figure A5 again appears.

From Figure A5, the next application is selected by choosing option 2, "Find safety reqs." Again this application is preceded by an introduction screen (Figure A12) which only appears once the first time this application is run.

The user first selects the type of work to be performed on the component selected during the previous application. In this example, Figure A13 presents two options for tank maintenance. The "weld repair" option is selected.

Figures A14 and A15 display the Expert's response. Figure A14 describes the quality controls, Skill level, and permission requirements necessary, and Figure A15 lists unique safety precautions which must be followed while performing the specific maintenance task. Figure A16 allows the user to run this consultation again for a different type of work, or return to the meta application.

After returning to the meta application, Figure A5 again appears allowing the user to select the final application in this session by choosing option 3, "Get impact on ship." Figure A17 shows the initial introduction screen presented the first time the Impact application is run. The user then inputs the expected duration of the maintenance by selecting one of the options as shown in Figure A18. In this example, "2 days" is chosen. Finally, Figure A19 shows the Expert response on how the given maintenance task will affect other departments on the ship. The user can either run the application again for a different job duration or return to the meta program and again be presented with a screen as shown in Figure A5. Here the user can start a new session, or exit out of the meta application. To end the session, the user selects option 4, "End consultations" and exits to the *VP-Expert* opening screen (Figure A1). To return to DOS, the user selects option 8, "Quit."

```
V P - E X P E R T
Version 2.02
Copyright (c) 1988
Brian Sawyer
All Rights Reserved

Editor Portion Copyright (c) 1984, 1985, 1987, Idea Ware Inc.

Published by Paperback Software International




1Help   2Induce  3Edit   4Consult 5Tree   6FileName 7Path   8Quit
1Help  2Go  3Whatif  4Variable  5Rule  7Set  8Quit
```

Figure A1 *VP_Expert's* opening screen

```
V P - E X P E R T
Version 2.02
Copyright (c) 1988
Brian Sawyer
All Rights Reserved

Editor Portion Copyright (c) 1984, 1985, 1987, Idea Ware Inc.

Published by Paperback Software International

What is the name of the knowledge base you want to use?

IMPACT    META    SAFETY    SYSTEMS
```

Figure A2 *VP_Expert's* screen listing available knowledge bases

The

SUBMARINE

MAINTENANCE

EXPERT

Written by LT David ACTON, USN

Naval Postgraduate School
Monterey, California
Version 1.0

Figure A3 The Shipboard Duty Officer's introduction screen

Welcome to the Submarine Maintenance Expert, an expert system designed specifically for the inport Submarine Duty Officer. This meta-Expert System is designed to show how VP Expert can share data with several distributed ES applications to provide important recommendations for the SDO in a variety of maintenance related areas.

The actual databases used by the Expert Systems contained in this first release are designed to show the capabilities of VP Expert, and thus in no way represent actual US Navy shipboard policies. However, only the database data used by each Expert System needs to be modified to implement this on board ship. The individual Expert System designs are complete.

Press any key to begin the consultation...

Figure A4 Author's welcoming screen and disclaimer

```
To fully utilize this Expert System, run this application
in the listed order, or select '4' to end all consultations.

1 Select component     2 Find safety reqs
3 Get impact on ship   4 End consultations
```

Figure A5 The Shipboard Duty Officer's application selection screen

```
  The maintenance Database program.  This application simply
allows the user to localize the exact component to be repaired,
and then stores data in a temporary file for use by other expert
systems.
  This application uses SYSTEMS.DBF and META for data.

  This is application #2 for LT David ACTON's thesis.

       Press any key to continue...
```

Figure A6 Component application's introduction screen

```
  On which system will maintenance be performed?

main seawater          hydraulics            H.P. Air
sonar                  sanitary              ventilation
chilled water          diesel fuel oil       trim and drain
electrical dist.
```

Figure A7 Screen to select system undergoing maintenance

```
  Which of the following best describes the general type
of component to be repaired?

pump                   pump motor            tank
valve                  nash float valve
```

Figure A8 Screen to select component type

In what area of the ship is the component to be repaired?

torpedo room MCLL ERLL

Figure A9 Screen to select shipboard maintenance area

What specific component is to undergo maintenance?

forward trim tank aux. tank #2

Figure A10 Screen to select specific component

```
   What specific component is to undergo maintenance?

forward trim tank     aux. tank #2



   You have selected aux. tank #2 as the item to undergo
maintenance.  Do you wish to move on to the consultations or
select a different component?


choice OK move on     select new component
```

Figure A11 Confirmation of selected component, or option to run application again

```
   The maintenance Safety Expert System  This expert system will
inform the Shipboard Duty Officer of special safety requirements
for various maintenance items.
   This ES calls on SYSTEMS.DBF and META for data.

   This is application #3 for LT David ACTON's thesis.

                  Press any key to continue...
```

Figure A12 Safety application's introduction screen

```
   What best describes the work to be performed on the
aux. tank #2?

clean and inspect    weld repair
```

Figure A13 Work options for selected component

```
   To weld repair the aux. tank #2, be aware that:

1. The job is not SUBSAFE.
2. It is not within ship's force capabilities.
3. The ENG must approve the tagout, and
4. The CO must give permission to start work.


   Press any key to continue this consultation...
```

Figure A14 Safety application's Expert requirements response

In addition, to weld repair the aux. tank #2, be aware of the following safety considerations:

- Possibly unsafe breathing environment:
Ensure aux. tank #2 is adequately ventilated and certified as safe to enter by a gas-free engineer, or ensure proper air-fed breathing equipment is used.

Press any key to continue...

Figure A15 Safety application's Expert safety response

Do you wish to run this consultation again for a different type of work on the aux. tank #2, or would you rather return to the META application?

run this again   **return to META**

Figure A16 Safety application's continuation or exit screen

```
   The maintenance Impact Expert System.  This expert system
will inform the Shipboard Duty Officer how maintenance will affect
other departments on the ship.
   This ES calls on IMPACT.DBF and META for data.

   This is ES application #4 for LT David ACTON's thesis.

      Press any key to continue...
```

Figure A17 Impact application's introduction screen

```
   How long is the maintenance task on the trim and drain system
in torpedo room expected to take?

less than 1 hour          1-4 hours                 4-12 hours
12-24 hours               2 days                    3-7 days
2 weeks                   more than 2 weeks
```

Figure A18 Screen to select expected maintenance duration

Due to the maintenance on the trim and drain system in torpedo room taking 2 days, the following department heads need to be notified for the reasons indicated:

- Contact the Weapons Officer and the CO if any welding operations are to take place in either the Missile Compartment or the Torpedo Room. In the Missile Compartment, weapons must be off-loaded from each tube neighboring the welding site. In the Torpedo Room, every torpedo must be off-loaded prior to beginning any hot work.

- none (no major shipwide impact).

Would you care to run this consultation for the trim and drain system in torpedo room again, or would you like to end this consultation and return to the META application?

run this again   **return to META**

Figure A19 Impact application's Expert response

APPENDIX B

I. *VP-EXPERT* APPLICATIONS

A. OVERVIEW

Each of the four applications used by the Submarine Maintenance Expert were written using *VP-Expert*. Code definitions can be found in reference 10.

B. META APPLICATION

```
!                          META.KBS
!
!     LT David ACTON                Thesis application #1
!
!     The Submarine Maintenance Meta-Expert System
!
!    This application calls on SYSTEMS.KBS, SAFETY.KBS,
!    IMPACT.KBS, and META for all data
!***********************************************************

RUNTIME;
EXECUTE;
ACTIONS

!  Find out if this consultation has been run before by
!  checking the text file "META."  If this is the first time
!  run, META should contain "do_again = start CNF 100"


LOADFACTS META
WHILETRUE do_again = start THEN
   WOPEN 1,1,2,20,75,1
   WOPEN 2,2,11,18,54,5
   ACTIVE 2
   COLOR = 0

   DISPLAY       "
```

```
            Written by LT David ACTON, USN

               Naval Postgraduate School
                 Monterey, California
                     Version 1.0 "


      WOPEN 3,3,30,8,16,7
      ACTIVE 3
      COLOR = 4
      DISPLAY "
       The

    SUBMARINE

   MAINTENANCE

     EXPERT ~"

      WCLOSE 3
      WCLOSE 2
      WOPEN 4,2,3,17,73,8
      ACTIVE 4
      COLOR = 15
      DISPLAY "
     Welcome to the Submarine Maintenance Expert, an expert
 system designed specifically for the inport Submarine Duty
 Officer.  This meta-Expert System is designed to show how VP
 Expert can share data with several distributed ES applica-
 tions to provide important recommendations for
 the SDO in a variety of maintenance related areas.

     The actual databases used by the Expert Systems con-
 tained in this first release are designed to show the capab-
 ilities of VP Expert, and thus in no way represent actual US
 Navy shipboard policies.  However, only the database data
 used by each Expert System needs to be
 modified to implement this on board ship.  The individual
 Expert System designs are complete.

     Press any key to begin the consultation...~"
      WCLOSE 4
      do_again = meta
 END
```

```
!*********************** End of Introduction ***************

CLS

WHILETRUE do_again = meta THEN

         WOPEN 5,3,4,14,72,1
         WOPEN 6,4,6,12,68,8
         ACTIVE 6
         COLOR = 15
         RESET chain_to_call
         RESET chain_called
         FIND chain_called

END

SAVEFACTS META;

!**************************** Rules Block ******************

RULE 0
IF       chain_to_call = 1_Select_component
THEN     chain_called = true
    CHAIN systems;


RULE 1
IF       chain_to_call = 2_Find_safety_reqs
THEN     chain_called = true
    CHAIN safety;


RULE 2
IF       chain_to_call = 3_Get_impact_on_ship
THEN     chain_called = true
    CHAIN impact;


RULE end
IF   chain_to_call = 4_End_consultations
THEN chain_called = end
     RESET ALL
     do_again = start;

!***************** Get input from user Block ***************

ASK chain_to_call: "
   To fully utilize this Expert System, run this application
 in the listed order, or select '4' to end all consulta-
tions.
```

```
";

CHOICES chain_to_call: 1_Select_component,
2_Find_safety_reqs,
3_Get_impact_on_ship, 4_End_consultations;
```

C. SYSTEMS APPLICATION

```
!                            SYSTEMS.KBS
!
!     LT David ACTON                 Thesis application #2
!
!     A Shipboard Duty Officer's Maintenance related
!                       Database Program
!
!    This application uses SYSTEMS.DBF and META for all data
!
!************************************************************
!    Initial screen and intro

RUNTIME;
EXECUTE;
ACTIONS


     DISPLAY "
    The Maintenance Database program.  This application
simply allows the user to localize the exact component to be
repaired, and then stores data in a temporary file for use
by other expert systems.
    This application uses SYSTEMS.DBF and META for data.

    This is application #2 for LT David ACTON's thesis.

            Press any key to continue...~"

!************************************************************
!    Check to see if user wants to run this consultation
!    again, if so, run it without showing above intro screen

do_again = yes
WHILETRUE do_again = yes THEN

    CLS

!    Find out on which system the maintenance is to be
!    performed

     RESET ALL
```

```
     CLOSE SYSTEMS
     MENU VP_system, ALL, SYSTEMS, system
     RESET VP_system
     FIND VP_system
     MRESET VP_system


!************************************************************
!      Find out the general description of the maintenance

      CLS
      MENU VP_description, VP_system = system, SYSTEMS,
descriptn
      RESET VP_description
      FIND VP_description
      MRESET VP_description


!************************************************************
!        Find the location on the ship where the maintenance
!     is to be performed

      CLS
      MENU VP_location, VP_system = system AND
VP_description = descriptn,
      SYSTEMS, location
      RESET VP_location
      FIND VP_location
      MRESET VP_location


!************************************************************
!        Now localize the exact component to be repaired

      CLS
      MENU VP_component, VP_system = system AND
VP_description = descriptn
      AND VP_location = location, SYSTEMS, component
      RESET VP_component
      FIND VP_component
      MRESET VP_component


    RESET continue_systems
    RESET do_again
    FIND do_again

END
```

```
do_again = meta
SAVEFACTS META
CHAIN META;       ! Return to the META application


!************************* Rules Block *********************

RULE 0
IF continue_systems = select_new_component
THEN do_again = yes
ELSE do_again = no;

!***************** Get input from user Block ***************


ASK VP_system: "
    On which system will maintenance be performed?
";

ASK VP_description: "
    Which of the following best describes the general type
of component to be repaired?
";

ASK VP_location: "
    In what area of the ship is the component to be
repaired?
";

ASK VP_component: "
    What specific component is to undergo maintenance?
";

ASK continue_systems: "

    You have selected {VP_component} as the item to undergo
maintenance.
 Do you wish to move on to the consultations or select a
different component?
";
CHOICES continue_systems: choice_OK_move_on,
select_new_component;
```

D. SAFETY APPLICATION

```
!                    SAFETY.KBS
!
!      LT David ACTON              Thesis ES application #3
```

```
!
!    A Submarine Maintenance Safety Expert System
!
!    This program uses SAFETY.DBF and META for all data
!
!************************************************************
!    Initial screen and intro


RUNTIME;
EXECUTE;
ACTIONS


     DISPLAY "
    The Maintenance Safety Expert System.  This expert
system will inform the user of special safety requirements
for various maintenance items.
   This ES calls on SYSTEMS.DBF and META for data.

   This is application #3 for LT David ACTON's thesis.

                           Press any key to continue...~"

!************************************************************

!  If this application is to be run again, don't show intro
!  screen


LOADFACTS META
do_again = yes
WHILETRUE do_again = yes THEN

      CLS
      CLOSE SAFET`
      MENU VP_woı  , VP_component = component, SAFETY, work
      RESET VP_work
      FIND VP_work
      MRESET VP_work

!************************************************************
!    Provide the expert response


        CLS
        CLOSE SAFETY
        GET VP_component = component AND VP_work = work,
SAFETY, ALL
```

```
                RESET Is_subsafe
                FIND Is_subsafe

                RESET Is_shipforce
                FIND Is_shipforce

                DISPLAY "

         To {work} the {component}, be aware that:

      1.       The job {Is_subsafe} SUBSAFE.
      2.       It {Is_shipforce} within ship's force capabilities.
      3.       The {tagout} must approve the tagout, and
      4.       The {permission} must give permission to start
    work.


          Press any key to continue this consultation...~"



          CLS


          DISPLAY "

         In addition, to {work} the {component},
      be aware of the following safety considerations:"

                 RESET Is_electrical
                 FIND Is_electrical

                 RESET Is_flooding
                 FIND Is_flooding

                 RESET Is_radcon
                 FIND Is_radcon

                 RESET Is_no_smoking
                 FIND Is_no_smoking

                 RESET Is_ventilate
                 FIND Is_ventilate

                 RESET Is_none
                 FIND Is_none

          DISPLAY"
```

```
     Press any key to continue...~"
   CLS
   RESET continue_safety
   RESET do_again
   FIND do_again
END


do_again = meta
SAVEFACTS META
CHAIN META;                  ! Return to the META application


!**************** End of Action Block **********************

RULE 0
IF subsafe = T OR subsafe = Y
THEN Is_subsafe = is
ELSE Is_subsafe = is_not;

RULE 1
IF shipforce = T OR shipforce = Y
THEN Is_shipforce = is
ELSE Is_shipforce = is_not;


RULE 2
IF electrical = T OR electrical = Y
THEN Is_electrical = is
     DISPLAY "
      - Working in vicinity of energized equipment:
 Ensure all appropriate electrical safety precaution of
 NAVSEA 5000.1 are strictly followed.";


RULE 3
IF flooding = T OR flooding = Y
THEN Is_flooding = is
     DISPLAY "
      - Flooding possibility exists:
 Ensure that there are at least two methods of dewatering
 the ship in the vicinity of {VP_location}.";


RULE 4
IF radcon = T OR radcon = Y
THEN Is_radcon = is
     DISPLAY "
      - Possibility of radioactive contamination exists:
```

```
Observe all applicable RADCON Manual precautions to minimize
personnel exposure and to prevent the spread of contamina-
tion.";


RULE 5
IF no_smoking = T OR no_smoking = Y
THEN Is_no_smoking = is
     DISPLAY "
      - Explosive atmosphere/unsafe gasses in environment:
 Ensure the smoking lamp is out in vicinity of
{VP_location}.";


RULE 6
IF ventilate = T OR ventilate = Y
THEN Is_ventilate = is
     DISPLAY "
      - Possibly unsafe breathing environment:
 Ensure {component} is adequately ventilated and certified
 as safe to enter by a gas-free engineer, or ensure proper
 air-fed breathing equipment is used.";


RULE 7
IF Is_electrical <> is AND Is_flooding <> is AND
   Is_radcon <> is AND Is_no_smoking <> is AND
   Is_ventilate <> is
THEN Is_none = true
     DISPLAY "
              none.
";


RULE 8
IF continue_safety = run_this_again
THEN do_again = yes
ELSE do_again = no;
!*************** Get input from user Block ***************

ASK VP_work:"
        What best describes the work to be performed on the
 {VP_component}?
";

ASK continue_safety:"
        Do you wish to run this consultation again for a
 different type of work on the {component}, or would
 you rather return to the META application?
";
```

```
CHOICES continue_safety: run_this_again, return_to_META;
```

E. IMPACT APPLICATION

```
!                         IMPACT.KBS
!
!      LT David ACTON              Thesis ES application #4
!
!         A Shipboard Duty Officer's Maintenance Expert
!     System to determine shipwide impact of a particular
!     maintenance evolution.
!
!     This application uses IMPACT.DBF and META for all data
!
!************************************************************
!     Initial screen and intro


RUNTIME;
EXECUTE;
ACTIONS



     DISPLAY "
    The Maintenance Impact Expert System.  This expert
system will inform the user how maintenance will affect
other departments on the ship.
    This ES calls on IMPACT.DBF and META for data.

    This is ES application #4 for LT David ACTON's thesis.

         Press any key to continue...~"


!************************************************************

!  Load the variables from the previous applications.  This
!  consultation needs VP_system and VP_location

LOADFACTS META

!  If this application is run again, don't show initial
!  intro screen

do_again = yes
WHILETRUE do_again = yes THEN
```

```
!  Find the expected duration of the maintenance task.

   CLS

        RESET VP_duration
        FIND VP_duration

!  Provide the Expert System response for each impact area
!  (database field value = "true").

   CLS
   DISPLAY "
   Due to the maintenance on the {VP_system} system in
 {VP_location} taking {VP_duration}, the following depart-
ment heads need to be notified for the reasons indicated: "

   CLOSE IMPACT          !  Reset database pointers
        GET VP_system = system AND VP_location = location,
IMPACT, ALL

       RESET galley_comments
       FIND galley_comments

       RESET navigation_comments
       FIND navigation_comments

       RESET rx_safety_comments
       FIND rx_safety_comments

       RESET welding_warning
       FIND welding_warning

       RESET computer_comments
       FIND computer_comments

       RESET no_impact
       FIND no_impact

!  Expert system response complete, check to see if user
!  desires to run consultation again

        RESET continue_impact
        RESET do_again
        FIND do_again

END

CLS

do_again = meta
```

```
SAVEFACTS META
CHAIN meta;     !  Return to META application

!******************* Rules Block **************************

RULE 0
IF      VP_duration = 4-12_hours AND
        galley = T OR galley = Y
THEN    galley_comments = yes
   DISPLAY "
   - Inform the Supply Officer to minimize the use of galley
equipment and to try to keep the freezer and chillbox closed
for the duration of maintenance.";

RULE 1
IF      VP_duration = 12-24_hours AND
        galley = T OR galley = Y
THEN    galley_comments = yes
   DISPLAY"
   - Inform the Supply Officer that he should consider
shutting down the galley and will need to be extra careful
to keep the freezer/chillbox closed to prevent spoilage.
Additionally, cold meals should be served during the main-
tenance period.";

RULE 2
IF      VP_duration = 2_days AND
        galley = T OR galley = Y
THEN    galley_comments = yes
   DISPLAY "
   - Inform the Supply Officer that he must shut down the
galley and that he must lock the freezer/chillbox to prevent
any food spoilage due to the door being opened. Additional-
ly, personnel should be asked to eat meals off the ship.";

RULE 3
IF      VP_duration = 3-7_days AND
        galley = T OR galley = Y
THEN    galley_comments = yes
   DISPLAY "
   - Inform the Supply Officer that the galley must be shut
down and that the refrigerated goods must be off loaded to a
pier facility.  Additionally, the frozen goods should also
be off loaded to a pier facility.  Personnel must eat all
meals off ship.";

RULE 4
IF      VP_duration = 2_weeks OR VP_duration =
more_than_two_weeks AND
        galley = T OR galley = Y
```

```
THEN    galley_comments = yes
   DISPLAY "
   - Inform the Supply Officer that the galley must be shut
down and both the refrigerated goods and the frozen goods
must be off loaded to a pier facility.  Personnel must eat
all meals off ship. ";

RULE 5
IF      VP_duration = less_than_1_hour AND
        navigation = T OR navigation = Y
THEN    navigation_comments = yes
   DISPLAY "
   - Check with the Navigator to ensure no sensitive equip-
ment is being calibrated. ";


RULE 6
IF      VP_duration = 1-4_hours AND
        navigation = T OR navigation = Y
THEN    navigation_comments = yes
   DISPLAY "
   - Inform the Navigator.  He may wish to consider taking
some of the sensitive equipment off-line or provide a means
off temporary cooling.";

RULE 7
IF      VP_duration = 4-12_hours AND
        navigation = T OR navigation = Y
THEN    navigation_comments = yes
   DISPLAY "
   - Inform the Navigator.  He will probably need to install
a temporary means of cooling to his sensitive equipment. ";

RULE 8
IF      VP_duration = 12-24_hours AND
        navigation = T OR navigation = Y
THEN    navigation_comments = yes
   DISPLAY "
   - Inform the Navigator.  Temporary cooling must be in-
stalled to all sensitive equipment before maintenance
begins. ";

RULE 9
IF      VP_duration = 2_days AND
        navigation = T OR navigation = Y
THEN    navigation_comments = yes
   DISPLAY "
   - Inform the Navigator.  Sensitive equipment calibrations
may need to be coordinated with this maintenance.  Temporary
cooling must be installed to all sensitive equipment. ";
```

```
RULE 10
IF      VP_duration = 3-7_days OR VP_duration = 2_weeks AND
        navigation = T OR navigation = Y
THEN    navigation_comments = yes
   DISPLAY "
   - Inform the Navigator.  All sensitive equipment should
be shut down before maintenance begins.  Long term equipment
calibrations must be carefully coordinated. ";

RULE 11
IF      VP_duration = more_than_two_weeks AND
        navigation = T OR navigation = Y
THEN    navigation_comments = yes
   DISPLAY "
   - Inform the Navigator.  All sensitive equipment must be
shut down.  No long term calibrations of equipment will be
possible this refit. ";


RULE 12
IF      rx_safety = T OR rx_safety = Y
THEN    rx_safety_comments = yes
   DISPLAY "
   Inform the Engineer and the CO.  Regardless of the main-
tenance duration, very special requirements as specified in
the Reactor Plant Operating and Maintenance manuals must be
observed.  This type of maintenance requires the utmost
planning and control. ";



RULE 13
IF   weapons = T OR weapons = Y
THEN   welding_warning = yes
   DISPLAY "
   - Contact the Weapons Officer and the CO if any welding
operations are to take place in either the Missile Compart-
ment or the Torpedo Room. In the Missile Compartment,
weapons must be off-loaded from each tube neighboring the
welding site.  In the Torpedo Room, every torpedo must be
off-loaded prior to beginning any hot work. ";


RULE 14
IF      VP_duration = less_than_1_hour AND
        computer = T OR computer = Y
THEN    computer_comments = yes
   DISPLAY "
```

```
   - Check with the Tactical Systems Officer to ensure no
computers are being calibrated or tested.  ";


RULE 15
IF       VP_duration = 1-4_hours AND
         computer = T OR computer = Y
THEN     computer_comments = yes
   DISPLAY "
   - Inform the Tactical Systems Officer.  He may wish to
consider taking some of the computers off-line or provide a
means off temporary cooling. ";

RULE 16
IF       VP_duration = 4-12_hours AND
         computer = T OR computer = Y
THEN     computer_comments = yes
   DISPLAY "
   - Inform the Tactical Systems Officer.  He will probably
need to install a temporary means of cooling to his com-
puters. ";

RULE 17
IF       VP_duration = 12-24_hours AND
         computer = T OR computer = Y
THEN     computer_comments = yes
   DISPLAY "
   - Inform the Tactical Systems Officer.  Temporary cooling
must be installed to all computers before maintenance
begins. ";

RULE 18
IF       VP_duration = 2_days OR VP_duration = 3-7_days AND
         computer = T OR computer = Y
THEN     computer_comments = yes
   DISPLAY "
   - Inform the Tactical Systems Officer.  Computer testing
may need to be coordinated with this maintenance.  Temporary
cooling must be installed to all computers.  Computers may
need to be shut down. ";

RULE 19
IF       VP_duration = 2_weeks OR VP_duration =
more_than_two_weeks AND
         computer = T OR computer = Y
THEN     computer_comments = yes
   DISPLAY "
   - Inform the Tactical Systems Officer.  All computers
should be shut down before maintenance begins.  Long term
computer testing must be carefully coordinated. ";
```

```
RULE 20
IF   galley_comments <> yes AND
   navigation_comments <> yes AND
   rx_safety_comments <> yes AND
   weapons_comments <> yes AND
   computer_comments <> yes
THEN   no_impact = yes
   DISPLAY "
     - none (no major shipwide impact).";


RULE 21
IF continue_impact = run_this_again
THEN do_again = yes
ELSE do_again = no;

!************* Get input from user Block *******************

ASK VP_duration: "
    How long is the maintenance task on the {VP_system}
system in {VP_location} expected to take?
";

CHOICES VP_duration: less_than_1_hour, 1-4_hours,
4-12_hours, 12-24_hours, 2_days, 3-7_days, 2_weeks,
more_than_2_weeks;

ASK continue_impact: "
     Would you care to run this consultation for the
 {system} system in {location} again, or would you like to
end this consultation and return to the META application?
";

CHOICES continue_impact: run_this_again, return_to_META;
```

LIST OF REFERENCES

1. Bui, Tung, "An Architecture for Distributed Knowledge-Based GDSS," *Proceedings of the International Federation of Operations Research Society*, Athens, Greece, to appear.

2. Kroenke, D.M., and Dolan, K.A., *Database Processing: Fundamentals, Design, Implementation*, 3rd ed., p.571, Science Research Associates, 1988.

3. Jain, Hemant K., "A Comprehensive Model for the Design of Distributed Computer Systems," *IEEE Transactions on Software Engineering*, v.SE-13, no.10, p.1092, October 1987.

4. Summers, R.C., "Local Area Distributed Systems," *IBM Systems Journal*, v.28, no.2, p.228, 1989.

5. Bui, T., and Jarke, M., "Communications Requirements for Group Decision Support Systems," *Journal of Management Information Systems*, v.II, no.4, p.11, Spring 1986.

6. Page-Jones, Meilir, *The Practical Guide to Structured Systems Design*, 2d ed., pp. 35 & 73, Prentice-Hall Inc., 1988.

7. Atkinson, C., Moreton, T., and Natali, A., *Ada for Distributed Systems*, p.51, Cambridge University Press, 1988.

8. Scott, Michael L., "Language Support for Loosely Coupled Distributed Programs," *IEEE Transactions on Software Engineering*, v.SE-13, no.1, p.88, January 1987.

9. Bui T., and others, "Meta-Modeling for Multiple Knowledge Bases", *Proceedings of ORSA/TIMS*, St. Louis, MO., October 1987.

10. Paperback Software, *VP-Expert Rule-Based Expert System Development Tool*, p.ix, 1989.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Attn: Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Adjunct Professor Marty McCaffrey Code AS/MF Naval Postgraduate School Monterey, California 93943-5000	1
4.	Associate Professor Tung Xuan Bui Code AS/BD Naval Postgraduate School Monterey, California 93943-5000	1
5.	Assistant Professor Magdi Kamel Code AS/KA Naval Postgraduate School Monterey, California 93943-5000	1
6.	LT David W. Acton, USN c/o Fay Brown 1000 Union Street, #208 Seattle, Washington 98101	1